Standard 9: QuickSort (worst-case vs expected behavior)

I was incorrect in assuming that QuickSort's worst input expected runtime was the same as its worst input worst-case runtime. I believed this to be true due to us discussing during lecture that by using randomness, QuickSort can be made to have the expected runtime O(nlogn) every time. This is not applicable to this question because the worst input worst-case runtime is O(n^2) due to how randomized QuickSort works in its worst-case runtime. In the worst-case runtime, QuickSort ends up using the worst pivot, which means that the pivot will either be set to the greatest value or smallest value in the sorted array. Due to this, the expected runtime for the worst input using QuickSort is not the same as the worst-case runtime for the worst input.

I believe that I did not fully understand what the question was asking and applied information that did not pertain to this question. The expected runtime will be O(nlogn) based on any input, but the worst-case runtime for the worst input will be O(n^2). This is due to the array ending up being split into all individual size 1 arrays before being put back together. We cannot apply the expected runtime of O(nlogn) in this case because we are explicitly declaring that we want to know worst-case runtime of QuickSort for the worst input possible.

I now understand that the expected runtime of QuickSort will be different than the worst-case runtime. I also understand now why the worst-case worst input runtime of Quicksort is O(n^2).